

**IN THE CLAIMS:**

1. (Currently Amended) A computer-implemented method for invoking execution of a graphical program, the method comprising:

selecting a graphical program in response to user input, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program, wherein the graphical program comprises at least one input received by a first node of the graphical program and at least one output produced by a second node of the graphical program;

specifying a functional interface for the graphical program, wherein said specifying comprises mapping the at least one input and the at least one output to respective entry point function parameters, wherein the functional interface is specific to the graphical program;

creating a shared library comprising the graphical program, wherein the shared library includes an entry point function created according to the functional interface specified for the graphical program;

calling ~~an~~ the entry point function of the shared library, wherein the entry point function is associated with the graphical program, and wherein said calling the entry point function comprises providing values for the respective input parameters in accordance with said mapping;

the entry point function directly invoking the graphical program in response to said calling the entry point function;

the graphical program executing in response to said invoking the graphical program.

2 - 3. (Cancelled)

4. (Previously Presented) The method of claim 1,

wherein the graphical program comprises executable code included in the shared library.

5. (Previously Presented) The method of claim 1,  
wherein said calling the entry point function comprises passing one or more parameters to the entry point function;

wherein said entry point function invoking the graphical program comprises the entry point function passing the one or more parameters to the graphical program.

6. (Previously Presented) The method of claim 1, wherein the graphical program produces one or more output values, the method further comprising:

the graphical program returning the one or more output values to the entry point function;

the entry point function returning the one or more output values.

7. (Previously Presented) The method of claim 1, wherein said calling the entry point function comprises passing a parameter to the entry point function, the method further comprising:

the entry point function transforming the parameter into a format expected by the graphical program;

the entry point function passing the transformed parameter to the graphical program.

8. (Previously Presented) The method of claim 1, wherein said calling the entry point function comprises passing a parameter to the entry point function, the method further comprising:

the entry point function copying the parameter into a location expected by the graphical program.

9. (Original) The method of claim 1,

wherein said calling the entry point function is performed by a particular thread;

wherein the graphical program executes within the context of the particular thread.

10 - 11. (Cancelled)

12. (Previously Presented) The method of claim 1,  
wherein said creating the shared library comprises excluding a portion of the graphical program that is not necessary for execution.

13. (Original) The method of claim 12,  
wherein the portion of the graphical program that is not necessary for execution comprises information from the group consisting of:  
user interface display information and block diagram information.

14. (Original) The method of claim 1,  
wherein the shared library is one of:  
a Windows DLL, a Unix shared library, and a Macintosh code fragment.

15. (Original) The method of claim 1,  
wherein said calling the entry point function of the shared library is performed by a program created using a text-based programming language.

16. (Currently Amended) A computer-implemented method for creating a shared library including a graphical program, the method comprising:

selecting a graphical program in response to user input, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program, wherein the graphical program comprises at least one input received by a first node of the graphical program and at least one output produced by a second node of the graphical program;

specifying a functional interface for the graphical program, wherein said specifying comprises mapping the at least one input and the at least one output to

respective entry point function parameters, and wherein the functional interface is specific to the graphical program;

creating a shared library comprising the graphical program, wherein the shared library includes a function created according to the functional interface specified for the graphical program, and wherein the function takes values for the respective input parameters in accordance with said mapping as inputs;

wherein the graphical program is directly invocable via the functional interface.

17. (Original) The method of claim 16,

wherein the graphical program has associated inputs and outputs;

wherein said specifying a functional interface for the graphical program comprises a user specifying a mapping of the associated inputs and outputs to parameters in a function declaration.

18. (Original) The method of claim 16,

wherein the graphical program has associated inputs and outputs;

wherein said specifying a functional interface for the graphical program comprises automatically mapping the associated inputs and outputs to parameters in a function declaration.

19. (Original) The method of claim 16,

wherein, in response to being invoked by a program, the shared library function created according to the functional interface specified for the graphical program is operable to invoke the graphical program.

20. (Currently Amended) A system comprising:

a computer including a CPU and memory;

a shared library stored in the memory of the computer, wherein the shared library includes an entry point function associated with a graphical program, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program;

wherein, in response to a program calling the entry point function, the entry point function is operable to directly invoke the graphical program, wherein said calling the entry point function comprises providing values for the respective input parameters in accordance with said mapping.

21. (Cancelled)

22. (Original) The system of claim 20,  
wherein the graphical program comprises executable code included in the shared library.

23. (Previously Presented) The system of claim 20,  
wherein said calling the entry point function comprises passing one or more parameters to the entry point function;

wherein said entry point function invoking the graphical program comprises the entry point function passing the one or more parameters to the graphical program.

24. (Previously Presented) The system of claim 20,  
wherein the graphical program produces one or more output values;  
wherein the graphical program returns the one or more output values to the entry point function;

wherein the entry point function returns the one or more output values.

25. (Previously Presented) The system of claim 20,  
wherein said calling the entry point function comprises passing a parameter to the entry point function;

wherein the entry point function transforms the parameter into a format expected by the graphical program;

wherein the entry point function passes the transformed parameter to the graphical program.

26. (Previously Presented) The system of claim 20,  
wherein said calling the entry point function comprises passing a parameter to the entry point function;

wherein the entry point function copies the parameter into a location expected by the graphical program.

27. (Original) The system of claim 20,  
wherein said calling the entry point function is performed by a particular thread;  
wherein the graphical program executes within the context of the particular thread.

28. (Original) The system of claim 20,  
wherein the shared library is one of:  
a Windows DLL, a Unix shared library, and a Macintosh code fragment.

29. (Original) The system of claim 20,  
wherein said calling the entry point function of the shared library is performed by a program created using a text-based programming language.

30. (Currently Amended) A computer-implemented system for creating a shared library including a graphical program, the system comprising:

a computer including a CPU and memory;  
a graphical programming system stored in the memory;

wherein the graphical programming system is operable to:

select a graphical program in response to user input, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program, wherein the graphical program comprises at least one input received by a first node of the graphical program and at least one output produced by a second node of the graphical program;

specify a functional interface for the graphical program, wherein said specifying comprises mapping the at least one input and the at least one output to

respective entry point function parameters, and wherein the functional interface is specific to the graphical program;

create a shared library comprising the graphical program, wherein the shared library includes a function created according to the functional interface specified for the graphical program, and wherein the function takes values for the respective input parameters in accordance with said mapping as inputs;

wherein the graphical program is directly invocable via the functional interface.

31. (Original) The system of claim 30,

wherein the graphical program has associated inputs and outputs;

wherein said specifying a functional interface for the graphical program comprises a user specifying a mapping of the associated inputs and outputs to parameters in a function declaration.

32. (Original) The system of claim 30,

wherein the graphical program has associated inputs and outputs;

wherein said specifying a functional interface for the graphical program comprises automatically mapping the associated inputs and outputs to parameters in a function declaration.

33. (Currently Amended) The system of claim 30,

wherein, in response to being invoked by a program, the shared library function created according to the functional interface specified for the graphical program is operable to directly invoke the graphical program.

34. (Currently Amended) A memory medium comprising program instructions operable to:

specify a functional interface for a graphical program, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program, wherein the graphical program comprises at least one input received by a first node of the graphical program and at least one output produced by a

second node of the graphical program, wherein said specifying comprises mapping the at least one input and the at least one output to respective entry point function parameters, and wherein the functional interface is specific to the graphical program;

create a shared library comprising the graphical program, wherein the shared library includes an entry point function created according to the functional interface specified for the graphical program;

call an entry point function of a the shared library, wherein the entry point function is associated with the graphical program, and wherein said calling the entry point function comprises providing values for the respective input parameters in accordance with said mapping;

directly invoke the graphical program in response to said calling the entry point function, wherein the entry point function directly invokes the graphical program;

execute the graphical program in response to said invoking the graphical program.

35 - 36. (Cancelled)

37. (Previously Presented) The memory medium of claim 34,  
wherein the graphical program comprises executable code included in the shared library.

38. (Previously Presented) The memory medium of claim 34,  
wherein said calling the entry point function comprises passing one or more parameters to the entry point function;

wherein said invoking the graphical program comprises the entry point function passing the one or more parameters to the graphical program.

39. (Previously Presented) The memory medium of claim 34,  
wherein the graphical program produces one or more output values  
wherein the graphical program returns the one or more output values to the entry point function;

wherein the entry point function returns the one or more output values.



40. (Previously Presented) The memory medium of claim 34,  
wherein said calling the entry point function comprises passing a parameter to the  
entry point function;

wherein the entry point function transforms the parameter into a format expected  
by the graphical program;

wherein the entry point function passes the transformed parameter to the graphical  
program.

41. (Previously Presented) The memory medium of claim 34,  
wherein said calling the entry point function comprises passing a parameter to the  
entry point function;

wherein the entry point function copies the parameter into a location expected by  
the graphical program.

42. (Original) The memory medium of claim 34,  
wherein said calling the entry point function is performed by a particular thread;  
wherein the graphical program executes within the context of the particular  
thread.

43 - 44. (Cancelled)

45. (Previously Presented) The memory medium of claim 34,  
wherein said creating the shared library comprises excluding a portion of the  
graphical program that is not necessary for execution.

46. (Original) The memory medium of claim 45,  
wherein the portion of the graphical program that is not necessary for execution  
comprises information from the group consisting of:  
user interface display information and block diagram information.

47. (New) A memory medium comprising program instructions operable to:

specify a functional interface for a graphical program, wherein the graphical program comprises a plurality of connected nodes which visually indicate functionality of the graphical program, wherein the graphical program comprises at least one input received by a first node of the graphical program and at least one output produced by a second node of the graphical program, wherein said specifying comprises mapping the at least one input and the at least one output to respective entry point function parameters, and wherein the functional interface is specific to the graphical program; and

create a shared library comprising the graphical program, wherein the shared library includes an entry point function created according to the functional interface specified for the graphical program, wherein the entry point function takes values for the respective input parameters in accordance with said mapping as inputs, and wherein the entry point function is callable to directly invoke the graphical program.